

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

Requested Patent: JP6161745A

Title:

METHOD AND APPARATUS FOR VERIFYING THE ORDERLY PROCESSING OF DATA ;

Abstracted Patent: EP0598471, B1 ;

Publication Date: 1994-05-25 ;

Inventor(s): FUJIOKA SHUNTARO (JP); NABEKURA HIDEAKI (JP) ;

Applicant(s): FUJITSU LTD (JP) ;

Application Number: EP19930306610 19930820 ;

Priority Number(s): JP19920307461 19921118 ;

IPC Classification: G06F9/38 ; G06F11/08 ;

Equivalents:

AU4449893, AU670802, CA2105113, DE69323695D, DE69323695T, US5530815

#### ABSTRACT:

Asynchronous computation commands sent from a command control are held in a command queue. The executable command is selected and supplied to a pipelined asynchronous computing unit. A status area is ensured for every command held in the command queue and pipeline bits indicative of the state of progress of the execution of the command in the unit are stored. A queue verifier discriminates the pipeline bits and verifies that a plurality of pipeline stages do not exist in an entry of any single command and that a plurality of pipeline stages do not exist among the commands, thereby guaranteeing the correct and orderly execution of the commands.

AF



## 【特許請求の範囲】

【請求項1】 命令を順番に解読し同期演算命令と非同期演算命令とに分けて異なる実行先に送る命令転送過程と、

該命令転送過程で送られた同期演算命令を実行する同期演算過程と、

前記命令転送過程で送られた非同期演算命令を保持して実行を待つ命令待合せ過程と、

該命令待合せ過程で保持した命令の中から実行可能な命令を選択すると共に命令実行終了まで選択した命令を保持させる命令選択過程と、

該命令選択過程で選択された非同期命令を複数段階に分けて実行するパイプライン化された非同期演算過程と、

前記命令待合せ過程で保持されている命令ごとにステータス領域(20)を確保し、前記非同期演算過程での命令実行の進行状態を示す情報を前記ステータス領域(20)に格納する格納する状態監視過程と、

前記ステータス領域(20)を検証する検証過程と、を備えたことを特徴とするデータ処理装置の検証方法。

【請求項2】 請求項1記載のデータ処理装置の検証方法に於いて、前記状態監視過程は、前記非同期演算過程の各パイプライン・ステージごとにステータスビットの各格納領域を設けて命令実行を示すビットフラグをセットすることを特徴とするデータ処理装置の検証方法。

【請求項3】 請求項2記載のデータ処理装置の検証方法に於いて、前記検証過程は、前記命令待合せ過程で保持された命令ごとにステータス領域(20)のステータスビットをチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証方法。

【請求項4】 請求項2記載のデータ処理装置の検証方法に於いて、前記検証過程は、前記命令待合せ過程で保持された複数の命令間でステータス領域(20)のステータスビットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証方法。

【請求項5】 請求項2記載のデータ処理装置の検証方法に於いて、前記検証過程は、前記命令待合せ過程で保持された命令ごとにステータス領域(20)のステータスビットをチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証すると同時に、前記命令待合せ過程で保持された複数の命令間でステータス領域(20)のステータスビットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証する特徴とするデータ処理装置の検証方法。

【請求項6】 請求項3又は5記載のデータ処理装置の検証方法に於いて、前記検証過程は、1つの命令のステータス領域に複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴とするデータ

処理装置の検証装置。

【請求項7】 請求項4又は5記載のデータ処理装置の検証方法に於いて、前記検証過程は、複数の命令の同一ステージに複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴とするデータ処理装置の検証方法。

【請求項8】 請求項1乃至7記載のデータ処理装置の検証方法に於いて、前記命令制御過程は、浮動小数点命令を解読した場合に該命令を前記命令待合せ過程に送ることを特徴とするデータ処理装置の検証方法。

【請求項9】 請求項1乃至7記載のデータ処理装置の検証方法に於いて、前記命令選択過程は、浮動小数点命令を選択した場合に前記非同期演算過程による命令実行の終了まで該浮動小数点命令を前記命令待合せ過程に保持させることを特徴とするデータ処理装置の検証方法。

【請求項10】 命令を順番に解読し同期演算命令と非同期演算命令とに分けて異なる実行先に送る命令制御手段(10)と、

該命令制御手段(10)から送られた同期演算命令を実行する同期演算手段(12)と、

前記命令制御手段(10)から送られた非同期演算命令を保持して実行を待つ命令待合せ手段(14)と、

該命令待合せ手段(14)の中から実行可能な命令を選択すると共に命令実行終了まで前記命令待合せ手段(14)に保持させる命令選択手段(16)と、

該選択手段(16)で選択された非同期命令を複数段階に分けて並列的に実行するパイプライン化された非同期演算手段(18)と、

前記命令待合せ手段(14)に格納された保持された命令ごとにステータス領域(20)を確保し、前記非同期演算手段(18)に於ける命令実行の進行状態を示す情報を前記ステータス領域(20)に格納する状態監視手段(22)と、

前記ステータス領域(20)を検証する検証手段(24)と、を設けたことを特徴とするデータ処理装置の検証装置。

【請求項11】 請求項10記載のデータ処理装置に於いて、前記状態監視手段(22)は、前記非同期演算手段(18)の各パイプライン・ステージごとにステータスビットの各格納領域を設けて命令実行を示すビットフラグをセットすることを特徴とするデータ処理装置の検証装置。

【請求項12】 請求項11記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された命令ごとにステータス領域(20)のステータスビットをチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証装置。

【請求項13】 請求項11記載のデータ処理装置の検証

装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された複数の命令間でステータス領域(20)のステータスピットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証装置。

【請求項14】請求項11記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された命令ごとにステータス領域(20)のステータスピットをチェックし、同一命令の  
10 エントリには複数のパイプライン・ステージが存在しないことを検証すると同時に、前記命令待合せ手段(14)に保持された複数の命令間でステータス領域(20)のステータスピットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証する特徴とするデータ処理装置の検証装置。

【請求項15】請求項12又は14記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、1つの命令のステータス領域に複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴とするデータ処理装置の検証装置。  
20

【請求項16】請求項13又は14記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、複数の命令の同一ステージに複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴とするデータ処理装置の検証装置。

【請求項17】請求項10乃至16記載のデータ処理装置の検証装置に於いて、前記命令制御手段(10)は、浮動小数点命令を解説した場合に該命令を前記命令待合せ手段(14)に送ることを特徴とするデータ処理装置  
30 の検証装置。

【請求項18】請求項10乃至17記載のデータ処理装置の検証装置に於いて、前記命令選択手段(16)は、浮動小数点命令を選択した場合に前記非同期演算手段(18)による命令実行の終了まで該浮動小数点命令を前記命令待合せ手段(14)に保持させることを特徴とするデータ処理装置の検証装置。

【請求項19】請求項10乃至19記載のデータ処理装置の検証装置に於いて、前記命令制御手段(10)および同期演算手段(12)で中央処理装置(26)を構成し、前記命令待合せ手段(14)、命令選択手段(16)、非同期演算手段(18)、状態監視手段(22)および検証手段(24)で非同期演算制御装置(28)を構成したことを特徴とするデータ処理装置の検証装置。  
40

【請求項20】命令を保持して実行を待つ命令待合せ手段(14)と、  
該命令待合せ手段(14)の中から実行可能な命令を選択すると共に命令実行終了まで前記命令待合せ手段(14)に保持させる命令選択手段(16)と、  
50

該選択手段(16)で選択された命令を複数段階に分けて並列的に実行するパイプライン化された非同期演算手段(18)と、

前記命令待合せ手段(14)に格納された保持された命令ごとにステータス領域(20)を確保し、前記非同期演算手段(18)に於ける命令実行の進行状態を示す情報を前記ステータス領域(20)に格納する状態監視手段(22)と、

前記ステータス領域(20)を検証する検証手段(24)と、を設けたことを特徴とするデータ処理装置の検証装置。

【請求項21】請求項20記載のデータ処理装置に於いて、前記状態監視手段(22)は、前記非同期演算手段(18)の各パイプライン・ステージごとにステータスピットの各格納領域を設けて命令実行を示すビットフラグをセットすることを特徴とするデータ処理装置の検証装置。

【請求項22】請求項21記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された命令ごとにステータス領域(20)のステータスピットをチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証装置。

【請求項23】請求項21記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された複数の命令間でステータス領域(20)のステータスピットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証装置。

【請求項24】請求項21記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、前記命令待合せ手段(14)に保持された命令ごとにステータス領域(20)のステータスピットをチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証すると同時に、前記命令待合せ手段(14)に保持された複数の命令間でステータス領域(20)のステータスピットをチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証することを特徴とするデータ処理装置の検証装置。  
40

【請求項25】請求項22又は24記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、1つの命令のステータス領域に複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴とするデータ処理装置の検証装置。

【請求項26】請求項23又は24記載のデータ処理装置の検証装置に於いて、前記検証手段(24)は、複数の命令の同一ステージに複数のパイプラインビットが存在した場合に、演算制御の異常と判定することを特徴と  
50

するデータ処理装置の検証装置。

【請求項27】請求項20乃至26記載のデータ処理装置の検証装置に於いて、前記命令選択手段(16)は、浮動小数点命令を選択した場合に前記非同期演算手段(18)による命令実行の終了まで該浮動小数点命令を前記命令待合せ手段(14)に保持させることを特徴とするデータ処理装置の検証装置。

【請求項28】請求項20乃至27記載のデータ処理装置の検証装置に於いて、前記命令待合せ手段(14)、命令選択手段(16)、非同期演算手段(18)、状態監視手段(22)および検証手段(24)で非同期演算制御装置(28)を構成したことを特徴とするデータ処理装置の検証装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、命令キューに保持した命令を取り出して実行するデータ処理装置の検証装置に関し、特に命令が正常終了するまで命令キューに命令を保持するデータ処理装置の検証方法及び装置に関する。近年、CPU装置による命令実行の高速化に伴い、同期演算器で1マシンサイクルごとに命令(同期命令)を逐次実行するだけでなく、複数サイクルに亘る長い命令(非同期命令)では別の演算器として非同期演算制御装置を設け、CPUと並列に命令実行を行っている。

【0002】そのため、非同期演算制御装置を設けた分だけデータ処理装置の回路規模が大きくなっており、LSI故障、論理不良が問題になり、信頼性を向上させる必要が増大してきている。

【0003】

【従来の技術】図12は命令を逐次実行する整数演算器12を備えた中央処理装置(CPU)26に対し長い非同期命令を並列的に実行する非同期演算制御装置28を設けた従来のデータ処理装置を示したもので、中央処理装置26に設けた命令制御部10で命令を解釈し、同期命令であれば同期演算器12に送って実行させ、非同期命令であれば非同期演算制御装置28に送って実行させる。

【0004】非同期演算制御装置28には命令を待ち行列として保持する命令キュー14、実行が可能となった命令を命令キュー14から選択する制御部16、制御部16で命令キュー14から選択された命令の投入を受けて複数サイクルで命令を実行する非同期演算器として機能する非同期パイプライン演算器18が設けられる。命令キュー14は、図13に示すように、命令コード30、第1ソースレジスタ番号32、第2ソースレジスタ番号34、結果書込みレジスタ番号(デスティネーション・レジスタ番号)36で構成される。

【0005】中央処理装置26の命令制御部10から命令が発行されると、命令キュー14に一旦保持するキューイングが行われる。キューイング後に実行可能となつ

た命令が制御部16で選択され、命令キュー14から出されて非同期パイプライン演算器18に投入される。このとき命令保存要求があれば、非同期パイプライン演算器18で命令実行が正常終了するまで命令キュー14を保持され、命令保存要求がなければ命令キュー14から削除される。

【0006】

【発明が解決しようとする課題】しかしながら、演算処理を高速化するために非同期演算制御装置を設けた場合には、その分だけデータ処理装置の回路規模が大きくなり、LSI故障や論理不良の可能性が高くなる。即ち、LSI不良、故障、断線、論理不良等が起きて非同期演算制御装置28に設けた命令キュー14に基づく非同期演算のパイプライン制御が乱された場合、待ち行列として命令を保持している命令キュー14の内容が保証できなくなり、命令キューの順序に従って動作せずに命令の実行に矛盾を生じ、異常終了してしまう問題があった。本発明の目的は、命令キューに保持された非同期命令を実行する際の順序と動作を保証するようにしたデータ処理装置の検証方法及び装置を提供する。

【0007】本発明の他の目的は、命令キューに保持する命令内容に加えて非同期パイプライン演算器におけるステージの命令実行段階を示すパイプラインビットを設けて命令の順序と動作を検証するデータ処理装置の検証方法及び装置を提供する。本発明の他の目的は、命令キューに設けたパイプラインビットに基づき、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証して命令実行の順序と動作を保証するデータ処理装置の検証方法及び装置を提供する。

【0008】本発明の他の目的は、命令キューに設けたパイプラインビットに基づき、並列的に実行されている複数の命令間では、同一のパイプライン・ステージが存在しないことを検証して命令実行の順序と動作を保証するデータ処理装置の検証方法及び装置を提供する。

【0009】

【課題を解決するための手段】図1は本発明の原理説明図であり、装置構成を例にとっている。まず本発明は、中央処理装置(CPU装置)26と非同期演算制御装置28とから構成される。中央処理装置26には、命令を順番に解釈し同期演算命令と非同期演算命令とに分けて異なる実行先に送る命令制御手段10と、命令制御手段10から送られた同期演算命令を実行する同期演算手段12が設けられる。

【0010】非同期演算制御装置28には、命令制御手段10から送られた非同期演算命令を保持して実行を待つ命令待合せ手段(命令キュー)14と、命令待合せ手段14の中から実行可能な命令を選択すると共に命令実行終了まで命令待合せ手段14に保持させる命令選択手段16と、選択手段16で選択された非同期命令を複数段階に分けて並列的に実行するパイプライン化された非

同期演算手段18とが設けられる。

【0011】更に本発明にあっては、命令持合せ手段14に格納された保持された命令ごとにステータス領域20を確保し、非同期演算手段18における命令実行の進行状態を示す情報（パイプラインビット）をステータス領域20に格納する状態監視手段22と、ステータス領域20を検証する検証手段24を設けたことを特徴とする。

【0012】検証手段24は、命令持合せ手段14に保持された命令ごとにステータス領域20のステータスビット（パイプラインビット）をチェックし、同一命令のエントリには複数のパイプライン・ステージが存在しないことを検証して命令実行の順序と動作を保証する。同時に検証手段24は、命令持合せ手段14に保持された複数の命令間でステータス領域20のステータスビット（パイプラインビット）をチェックし、各命令間には複数の同一パイプライン・ステージが存在しないことを検証して命令実行の順序と動作を保証する。

【0013】

【作用】このような構成を備えた本発明によるデータ処理装置の検証方法及び装置によれば、非同期演算制御装置の命令キュー14にパイプラインビットを格納するステータス領域20を新設することにより、次の条件を検証して命令実行の順序と動作を保証することができる。

【0014】条件1；命令キュー14の1つの命令のエントリ内には、複数のパイプライン・ステージが存在しないこと。

条件2；命令キュー14の各命令間では、複数の同一パイプライン・ステージが存在しないこと。

【0015】このためLSI不良、故障、断線、論理不良等が起きて非同期演算制御装置28に設けた命令キュー14に基づく非同期演算のパイプライン制御が乱された場合には、1つの命令のエントリにパイプラインビットが2以上立ったり、複数の命令間の同一ステージにパイプラインビットが2以上立つことから、命令実行の順序と動作の異常を直ちに認識し、適切なエラー回復処理に入ることができる。

【0016】このように命令キュー14パイプラインビットを追加して命令キューに基づく命令実行の順序と動作をチェックすることで命令キューの内容を保証できるようになる。

【0017】

【実施例】図2は本発明のデータ処理装置の全体構成を示した説明図であり、主記憶装置（MSU）100、主記憶制御装置（MCU）200及びプロセッサユニット300で構成される。プロセッサユニット300には、逐次、命令を実行する同期演算器としての機能を備えた中央処理装置26と、長い命令を非同期に実行するパイプライン化された演算器を備える非同期演算制御装置28が設けられる。

【0018】このような中央処理装置26と非同期演算制御装置28により並列的に命令を実行する高速処理を目的とした情報処理装置は、並列マシン構成のスーパーコンピュータに使用する各プロセッサエレメントのスカルユニットとして使用される。図3は図2のプロセッサユニット300に設けた中央処理装置26及び非同期演算制御装置28の詳細を示した実施例構成図である。

【0019】図3において、まず中央処理装置（CPU）26には命令制御部10、同期演算器としての機能する整数演算器12、命令実行に使用する各種の汎用レジスタを備えた汎用レジスタ群38、キャッシュ等のローカルメモリとして使用されるメモリ40が設けられる。命令制御部10は命令を順番に解釈し、同期演算命令を解釈すると命令を整数演算器12に送って、1サイクルで終了する整数演算を行わせる。また、命令制御部10で非同期演算命令、例えば浮動小数点演算命令を解釈した場合には、複数サイクルで実行される長い命令であることから、非同期演算制御装置28に送って並列的に命令を実行する。

【0020】非同期演算制御装置28には命令持合せ手段としての命令キュー14、制御部25、パイプライン化された非同期演算器としての非同期パイプライン演算器18が設けられる。本発明にあっては、命令キュー14は図4にインデックスで示される命令1～nごとに、命令コード30、第1ソースレジスタ番号32、第2ソースレジスタ番号34、結果レジスタ番号（デストネーション・レジスタ番号）36で構成され、更にステータス領域20としてパイプラインビットを設けている。

【0021】図3の非同期パイプライン演算器18にあっては、パイプラインがフェッチステージF、第1実行ステージE1、第2実行ステージE2及び書込ステージWの4ステージで構成されていることから、命令キュー14に保持した各命令のパイプラインビットとしてステージF、E1、E2、Wに対応した4つのパイプラインビットの格納領域を設けている。

【0022】再び図3を参照するに、非同期演算制御層28には制御部25が設けられ、制御部25には命令選択部16、状態監視部22及びキュー検証部24としての各機能が設けられる。命令選択部16は命令キュー14に待ち行列として保持された命令の中から実行可能となった命令を取り出して非同期パイプライン演算器18に投入する。

【0023】このとき、命令選択部18は命令キュー14から取り出した命令の第1ソースレジスタ番号32、第2ソースレジスタ番号34及び結果レジスタ番号36をレジスタアドレスバス44を通じて中央処理装置26の汎用レジスタ群38に通知し、対応するソースレジスタに格納しているソースデータをソースデータバス46により非同期パイプライン演算器18に供給する。

【0024】また非同期パイプライン演算器18によ

る実行結果を、レジスタアドレスバス44により指定された結果レジスタ番号で指定されるデストネーション・レジスタに書き込みデータバス48を通じて書き込むようにしている。更に、制御部25には命令キュー14のステータス領域20に設けたパイプラインビット(F, E1, E2, W)に対応して状態監視部22とキュー検証部24が設けられる。状態監視部22は非同期パイプライン演算器18に投入した命令の進行状況を監視し、フェッチステージ、第1実行ステージ、第2実行ステージ、書き込みステージに命令が移行する毎に、命令キュー10のステータス領域20に設けているパイプラインビットの書替えを行う。

【0025】図5は非同期パイプライン演算器18における命令の実行状態を示したタイムチャートである。図5において、まずT<sub>1</sub>サイクルで命令1が最初のフェッチステージに投入され、次のT<sub>2</sub>サイクルで第1実行ステージE1で第1ソースレジスタのソースデータに基づく処理を実行する。また、T<sub>3</sub>サイクルにあっては、次の命令2のフェッチステージFへの投入が行われる。

【0026】次のT<sub>4</sub>サイクルでは命令1は第2実行ステージE2に進み、第2ソースレジスタからのソースデータに基づく処理を実行し、命令2は第1実行ステージE1に進んで第1ソースレジスタのソースデータに基づく処理を実行し、更に命令3が新たにフェッチステージFに投入される。T<sub>5</sub>サイクルでは命令1は書き込みステージWに進んで実行結果をデストネーション・レジスタに書き込み、命令2は第2実行ステージE2に進み、更に命令3は第1実行ステージE1に進む。T<sub>6</sub>サイクルでは第2命令が書き込みステージW、第3命令が第2実行ステージE2となり、T<sub>7</sub>サイクルでは命令3が書き込みステージWとなる。

【0027】この図5に示すような複数命令の非同期パイプライン演算器18における並列的な実行状態は状態監視部22で監視され、命令キュー14のステータス領域20の各命令毎のパイプラインビット(F, E1, E2, W)が、対応する命令の存在するステージを示すパイプラインビットをビット1にセットし、それ以外のパイプラインビットを0にリセットしている。

【0028】制御部25に設けたキュー検証部24は非同期パイプライン演算器18の各サイクル毎に命令キュー14に保持されている複数の命令のステータス領域20をチェックし、命令キュー14に保持された命令に従った非同期パイプライン演算器18の順序及び動作が正常に行われているか否かを検証する。図6は、ある命令が非同期パイプライン演算器18に投入されて正常に実行された場合の命令キュー14のステータス領域20におけるパイプラインビットのサイクル変化を示している。これは、例えば図5に示した命令1のT<sub>1</sub>サイクルからT<sub>7</sub>サイクルまでの命令実行に対応している。尚、最初のT<sub>1</sub>サイクルは命令投入前のパイプラインビ

ットを示している。

【0029】このように正常に命令が実行された場合には、T<sub>1</sub>～T<sub>7</sub>サイクルの全てでパイプラインビット(F, E1, E2, W)のいずれか1つのビットのみが1となっている。即ち、同一命令のエントリ内には複数のパイプライン・ステージが存在しないことを示す。また命令投入前にあっては、パイプラインビットが全て0となっている。

【0030】本発明のキュー検証部24は命令キューのパイプラインビットについて各命令毎にNC1チェックを実行する。NC1チェックとは、4つのパイプラインビットがオール0かまたはどれか1つがビット1になった場合にチェック出力1を生じ、チェック出力1で正常動作が行われることを示す。一方、4つのパイプラインビットに2以上、ビット1があった場合には、チェック出力0を生じ、動作異常を示す。

【0031】図7は図6のT<sub>1</sub>サイクルでパイプラインビットFとE1がビット1となり、同一エントリに命令が2つ存在する動作異常を示しており、このT<sub>1</sub>サイクルでのNC1チェック結果は0となり、異常を起こしたことを検知することができる。図8はキュー検証部24における複数命令間のパイプラインビットの検証を示したもので、全ての命令において正常動作が行われている場合を示す。

【0032】図8にあっては、例えば命令番号1～5の5つの命令が命令キュー14に保持されており、それぞれパイプラインビット(F, E1, E2, W)に示すビット内容を持ち、正常動作が行われていることから、各命令間の同一ステージ、即ち縦方向に見たビット列の中でビット1は1つしか存在しない。即ち、キュー検証部24は複数命令間については同じステージに複数の命令が存在しないことで順序と動作を保証する。この場合にも、複数命令間の同一パイプラインビット毎にNC1チェックが行われ、正常動作であることから、命令間のチェック結果は全て1となる。

【0033】図9は図8の命令間のNCチェックにおいて異常が起きた場合を示しており、パイプラインビットE1について命令番号2と3にビット1が立っており、チェック結果が0となって動作異常が起きたことを認識することができる。本発明のキュー検証部24にあっては、図6、7に示した命令毎のNC1チェックと図8、9に示した命令間のNC1チェックのいずれか一方を行ってもよいが、どの命令の実行について異常が起きているかを判別するためには命令毎のNC1チェックと命令間のNC1チェックを同時に行うことが望ましい。

【0034】例えば、図9に示すような異常が起きている場合には、命令番号3の命令実行の順序と動作は正常であるが、次の命令番号4の動作と順序が誤っていることが認識でき、命令4のリトライで異常を回復することが可能となる。図10は図3に示した中央処理装置26



の命令制御部10による制御処理を示したフローチャートである。

【0035】図10において、まずステップS1で命令制御部10は所定の順番に従った命令を取り出し、ステップS2で命令を解説する。命令の解説結果からステップS3で浮動小数点演算命令が否かチェックし、整数演算命令であれば整数演算器12に送ってステップS4で整数演算命令を1サイクルで実行させる。一方、浮動小数点演算命令であればステップS8に進み、非同期演算制御装置28の命令キュー14に送る。

【0036】ステップS4で整数演算命令を実行すると、次のステップS5で非同期演算制御装置28に送った先行命令の終了待ちがあるか否かチェックし、もし非同期演算制御装置28に送った先行命令の終了待ちがあればステップS6に進み、非同期演算制御装置28からの先行命令の終了通知を待つ。ステップS5で終了待ちがなかった場合、あるいはステップS6で先行命令の終了通知を受けた場合にはステップS7に進み、命令番号をインクリメントし、再びステップS1に戻り、次の命令を取り出し、以下同様な処理を繰り返す。

【0037】図11は図3に示した非同期演算制御装置28に設けた制御部25による非同期制御を示したフローチャートである。図11において、まずステップS1で制御部25に設けた命令選択部16が実行可能となった命令を命令キュー14の待ち行列の中から選択して非同期パイプライン演算器18に投入する。

【0038】次にステップS2で、投入した命令について保存要求の有無をチェックし、もし保存要求がなければステップS8で投入済み命令を命令キュー14から削除する。本発明で検証の対象としている浮動小数点演算命令は、保存要求をもっていることから命令キュー14から削除されることなく、ステップS3に進み、非同期パイプライン演算器18で命令サイクルが実行される毎に命令が存在するパイプライン・ステージに対応するパイプラインビットを1にセットするビット更新を命令キュー14のステータス領域20に対し行う。

【0039】続いてステップS4でビット更新終了毎に命令キュー14のステータス領域20に格納しているパイプラインビットをチェックする。このチェックは命令毎のエントリ内のNC1チェック及び命令間のNC1チェックの両方を行う。ステップS5でNC1チェックのチェック結果から異常の有無を判定し、NC1チェックの結果が全て1であれば正常動作と判定し、ステップS6に進む。ステップS6にあっては、最終ステージを終了した命令があるか否かチェックし、最終ステージを終了した命令があればステップS7に進み、実行終了命令を命令キューから削除する。

【0040】一方、ステップS5でパイプラインビットのNC1チェックのチェック結果から0が得られた場合には異常ありと判定し、ステップS9に進み、エラー処

理を行う。例えば、異常を起こした命令を再度命令キュー14から取り出して非同期パイプライン演算器18に投入するリトライを行う。このリトライの実行によりステップS9でパイプラインビットのNC1チェックから正常動作となれば復旧が判定され、再びステップS1の処理に戻る。一方、異常の原因が断線や故障等のハードウェアに原因した場合には、リトライを行っても異常が回復しないことから、ステップS11に進んで異常終了とする。

10 【0041】この異常終了にあっては、例えば本発明のデータ処理装置が並列マシン構成のスーパーコンピュータのプロセッサエレメントのスカラユニットに使用されているような場合には、異常を起こしたスカラユニットをもつプロセッサエレメントを並列処理の対象から切り離す等の処理が行われることになる。尚、上記の実施例はスーパーコンピュータのプロセッサエレメントにおけるスカラユニットとして使用されるデータ処理装置を例にとっているが、本発明はこれに限定されず、適宜の計算機の中央処理装置としてそのまま使用することができる。

20 【0042】更に本発明は上記の実施例に限定されず、本発明の目的の範囲で可能な各種の変形を含むものである。

【0043】

【発明の効果】以上説明してきたように本発明によれば、LSI不良、故障、断線、論理不良等により、命令キューに基づく命令実行の制御に乱れが生じて異常が起きた場合、常に命令キューに設けているパイプラインビットをチェックしていることから、チェック結果に基づいて動作異常を認識し、リトライ、異常終了等の対応策を講ずることができ、常に命令キューを用いた命令実行の順序と動作を保証することができ、非同期演算制御装置の信頼性を大幅に向上することができる。

【図面の簡単な説明】

【図1】本発明の原理説明図

【図2】本発明の全体構成を示した説明図

【図3】本発明の実施例構成図

【図4】本発明で用いる命令キューの説明図

40 【図5】非同期パイプライン演算器における命令の並列実行を示したタイムチャート

【図6】正常に命令を実行した場合のパイプラインビットのチェックを示した説明図

【図7】命令実行中に異常を起した場合のパイプラインビットのチェックを示した説明図

【図8】正常動作時の各命令間のパイプラインビットのチェックを示した説明図

【図9】異常動作を起した時の各命令間のパイプラインビットのチェックを示した説明図

【図10】図3の中央処理装置の命令制御を示した説明

50 図

13

【図11】図3の非同期演算制御装置の非同期制御を示したフローチャート

【図12】従来のデータ処理装置の説明図

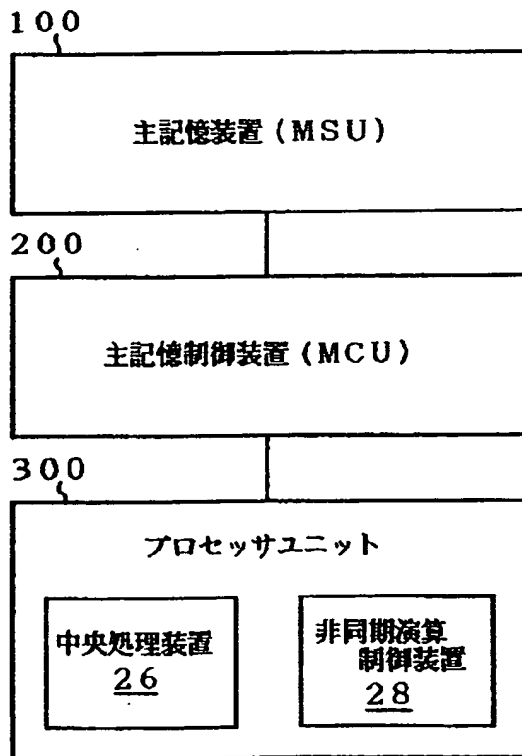
【図13】従来の命令キューの説明図

【符号の説明】

- 10：命令制御部（命令制御手段）  
 12：整数演算器（同期演算手段）  
 14：命令キュー（命令待合せ手段）  
 16：命令選択部（命令選択手段）  
 18：非同期パイプライン演算器（非同期演算手段）  
 20：ステータス領域  
 22：状態監視部（状態監視手段）  
 24：検証部（検証手段）  
 26：CPU（中央処理装置）

【図2】

本発明の全体構成を示した説明図



14

- 28：非同期演算制御装置  
 30：命令コード  
 32：第1ソースレジスタ番号  
 34：第2ソースレジスタ番号  
 36：結果レジスタ番号（ディストネーションレジスタ番号）  
 38：汎用レジスタ群  
 40：メモリ  
 44：レジスタアドレスバス  
 46：ソースデータバス  
 48：書込データバス  
 100：主記憶装置（MSU）  
 200：主記憶制御装置（MCU）  
 300：プロセッサユニット

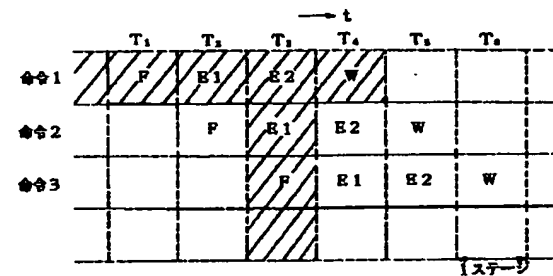
【図4】

本発明で用いる命令キューの説明図

命令コード 30	第1ソース レジスタ番号 32	第2ソース レジスタ番号 34	結果レジスタ 番号 36	パイプラインビット			
				F	E1	E2	W
命令5							
命令4							
命令3							
命令2							
命令1							

【図5】

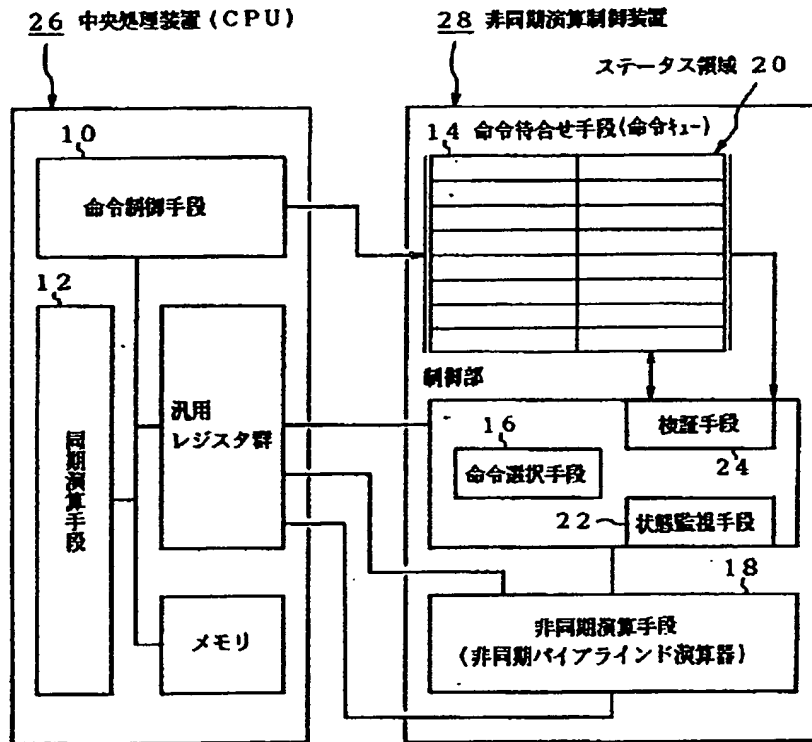
非同期パイプライン演算器における命令の並行実行を示したタイムチャート



【図1】

## 本発明の原理説明図

(a)

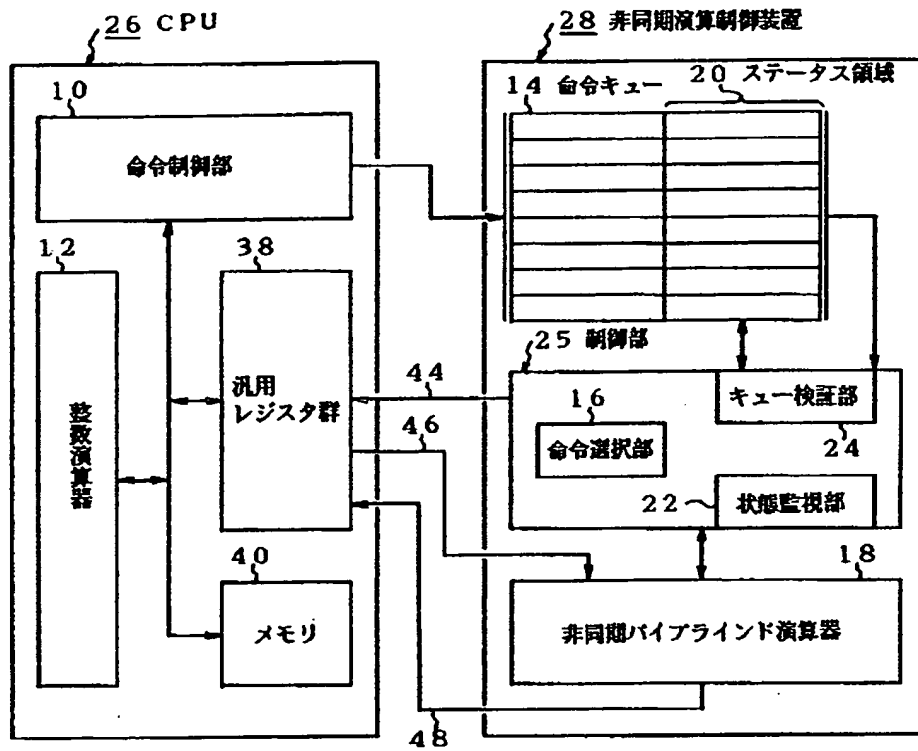


(b)

第1ソース 第2ソース 結果レジスタ パイプラインビット									
命令コード		レジスタ番号	レジスタ番号	番号	F	E1	E2	W	
命令n									
⋮									
命令2									
命令1									

【図3】

本発明の実施例構成図



【図6】

正常に命令を実行した場合のパイプラインビットのチェックを示した説明図

パイプラインビット

時間	F	I1	I2	W	パイプラインビット
T <sub>0</sub>	0	0	0	0	1
T <sub>1</sub>	1	0	0	0	1
T <sub>2</sub>	0	1	0	0	1
T <sub>3</sub>	0	0	1	0	1
T <sub>4</sub>	0	0	0	1	1

【図7】

命令実行中に異常を起した場合のパイプラインビットのチェックを示した説明図

パイプラインビット

時間	F	I1	I2	W	パイプラインビット
T <sub>0</sub>	0	0	0	0	1
T <sub>1</sub>	1	1	0	0	0 ←エラー
T <sub>2</sub>	0	1	0	0	1
T <sub>3</sub>	0	0	1	0	1
T <sub>4</sub>	0	0	0	1	1

【図8】

正常動作時の各命令間のパイプラインビットのチェックを示した説明図

パイプラインビット

命令番号	F	I1	I2	W
5	0	0	0	0
4	1	0	0	0
3	0	1	0	0
2	0	0	1	0
1	0	0	0	1
命令番号 フェッチ	1	1	1	1

【図9】

異常動作を起した時の各命令間のパイプラインビットのチェックを示した説明図

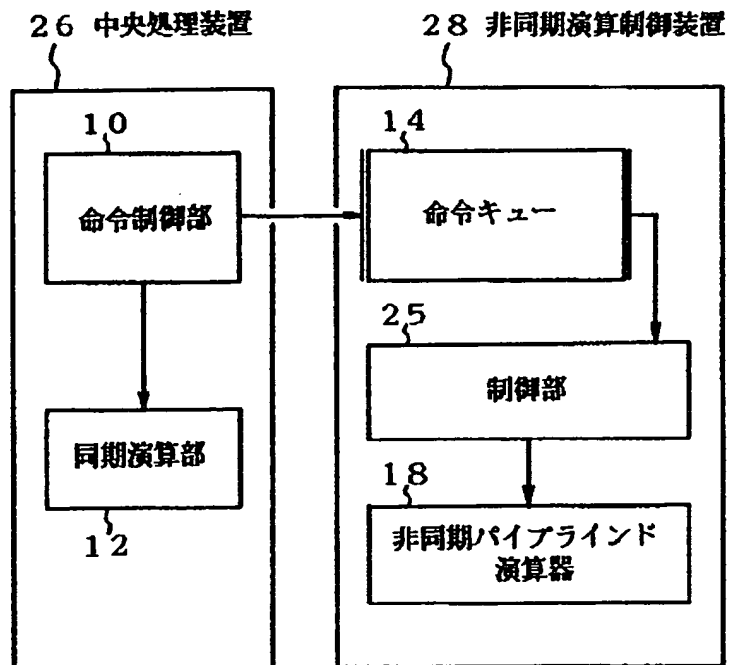
パイプラインビット

命令番号	F	I1	I2	W
5	0	0	0	0
4	1	1	0	0
3	0	1	0	0
2	0	0	1	0
1	0	0	0	1
命令番号 フェッチ	1	0	1	1

↑  
エラー

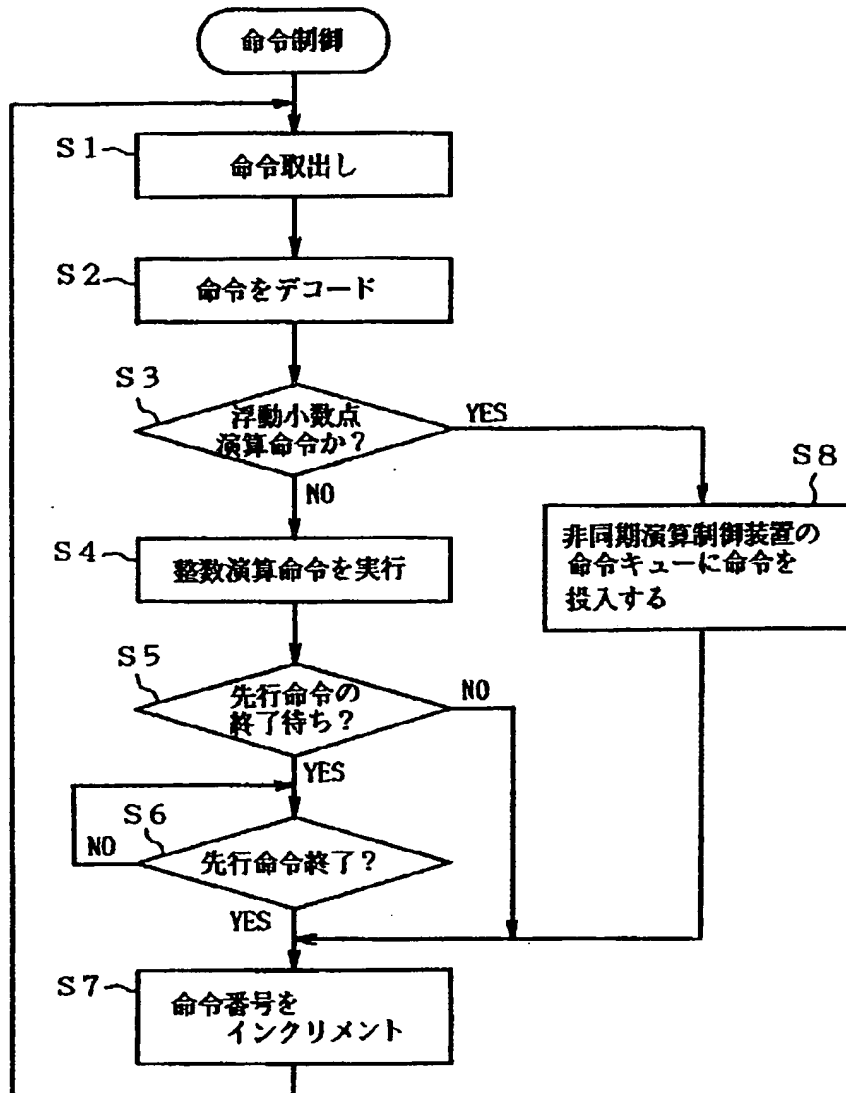
【図12】

従来のデータ処理装置の説明図



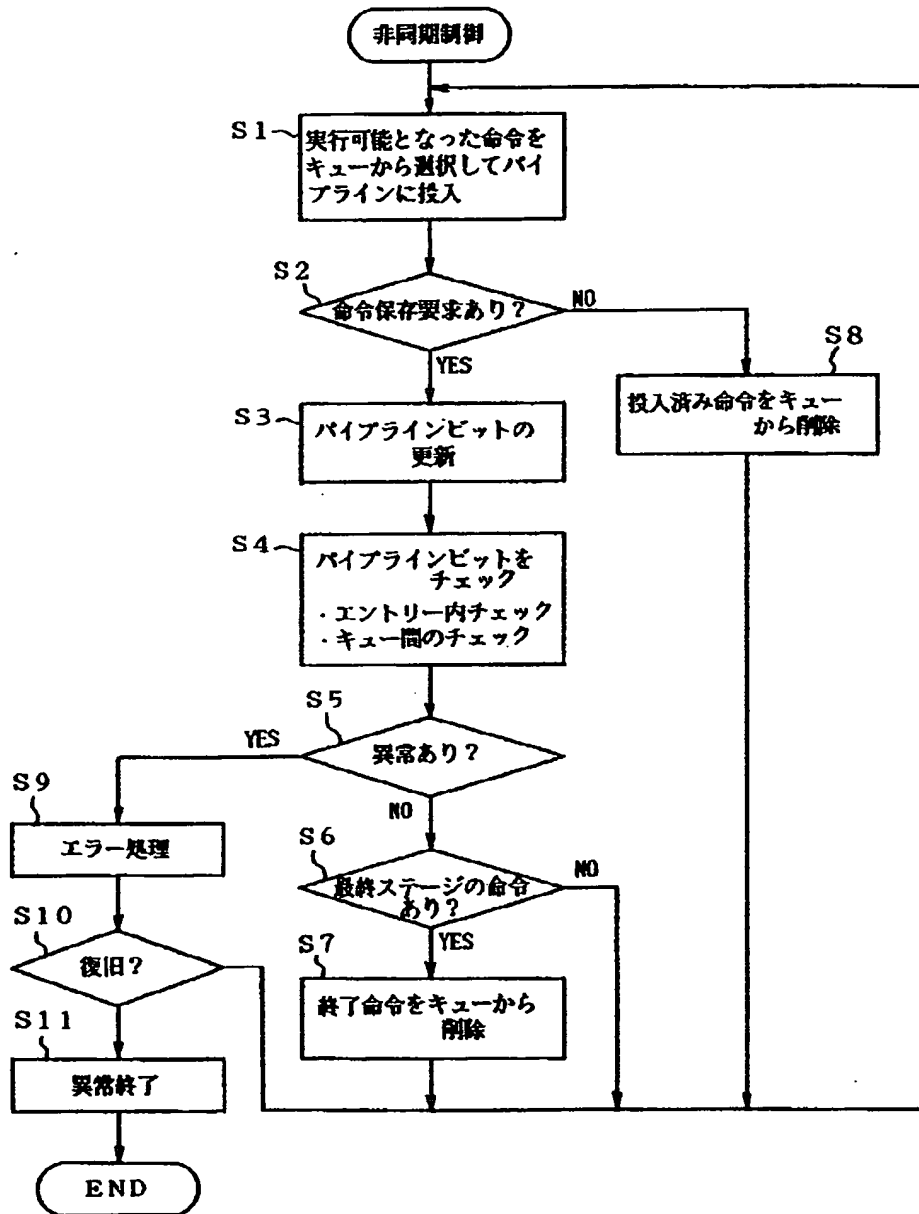
【図10】

図3の中央処理装置の命令制御を示した説明図



【図11】

図3の非同期演算制御装置の非同期制御を示したフローチャート



【図13】

## 従来の命令キューの説明図

	命令コード	第1ソース レジスタ番号	第2ソース レジスタ番号	結果レジスタ 番号
命令n				
⋮	⋮	⋮	⋮	⋮
命令2				
命令1				